

Session 5: Lessons Learned

Gerald Page, CSC, Discussant

Lorraine M. Duvall, School of Information Studies, Syracuse University

Sandra Yin and Julia McCreary, Internal Revenue Service

Paul Usavage, General Electric

INSIGHTS into SOFTWARE DEVELOPMENT in JAPAN

Lorraine M. Duvall
CASE Center and School of Information Studies
4-206 Center for Science and Technology
Syracuse University
Syracuse, NY 13244-4100
lmduvall@cat.syr.edu

Introduction

The interdependence of the US-Japanese economies makes it imperative that we in the United States understand how business and technology developments take place in Japan. We can gain insight into these developments in software engineering by studying the context in which Japanese software is developed, the practices that are used, the problems encountered, the setting surrounding these problems, and the resolution of these problems. Context includes the technological and sociological characteristics of the software development environment, the software processes applied, personnel involved in the development process, and the corporate and social culture surrounding the development.

Presented in this paper is a summary of the results of a study that addresses these issues. Data for this study was collected during a three month visit to Japan where the author interviewed 20 software managers representing nine companies involved in developing software in Japan.^{1 2} These data are compared to similar data from the United States in which 12 managers from five companies were interviewed.

The purpose of this field study is to increase our understanding of software problems and the situations in which these problems occur - from the perspective of software development managers. The collection and analysis of field data from actual software development settings has been successfully used for research that is problem-driven rather than technology-driven [CKI88, Sca89, SB88, Mye85]. The Computer Science and Technology Board of the National Research Council recommended that one method of performing research in software engineering is by studying software systems in corporate settings using methods from the behavioral and management sciences [CST89]. The methodology applied to this study is such an approach. Software managers were studied primarily in their own environment, using research methods from the social sciences.^{3 4 5}

This study uses qualitative research methods to explore the software development phenomenon from the perspective of the software manager, taking into account technical and

¹The interviews were conducted from December 13, 1991 to February 20, 1992.

²Two of the nine companies visited in Japan were affiliates of US multi-national corporations.

³Software managers are defined as second level managers, i.e., middle line managers who are two levels removed from the programmer.

⁴The research method was based upon grounded theory and the constant comparative method as applied to the social sciences [GS67, Pat90]. Using grounded theory, the results are sets of generalizations at varying levels of abstraction that are derived inductively from field data collected by the researcher.

⁵A modified form of open-ended interviewing was used to encourage the acquisition of a wide range of information from the software managers.

sociological factors. In contrast, most US-Japan comparative studies have addressed only technical characteristics of software development or were concerned with the state-of-the-art versus the state-of-practice. [HKG91, Cus89, Cus90, ZYH+84] [Mat89, Kim83, TM84, OB91]

The remainder of this paper includes discussions on the organization of the Japanese software industry, the state-of-practice of software development in Japan from the perspective of the managers interviewed, problems encountered, and comparisons to US practices and problems.

The Software Industry in Japan

The software industry in Japan is divided into four main groups: 1) computer and equipment manufacturers, 2) computer users, 3) subsidiaries of the two previous groups dedicated to software activities, and 4) independent software houses [Mat91, Mat89].

The first group, often-times referred to as "makers" or "mainframers", are the manufacturers of computer equipment. The computer users are the banks, security organizations, railroad and airline companies, automobile companies, etc. Both the computer manufacturers and the computer users have formed subsidiaries that are dedicated to software development to assist them in their development activities. The independent software houses perform systems analysis and development services, as do the subsidiaries of the computer manufacturers and the computer users.

Many of the computer users have little software development expertise in-house as they consider that in-house software development requires too much time and money and that outside purchasing of custom software development improves productivity [sof90]. Much of their software development work is done under contract through the computer manufacturers and the software houses, similar to the computer industry in the United States in the 1960s. There is much subcontracting involving all four groups.

Interview Profile

The interviews with software managers were planned to provide the coverage of a multitude of contexts in which software was developed. This included finding managers who were responsible for developing software for a range of application types (such as real-time control systems, information systems, software tools, office products) and different project sizes. Also of interest were the use of various hardware and software processes and tools used for developing the software.

The relationships between social and cultural contextual variables are of concern to this study. Therefore, managers working for different types of companies (such as computer manufacturer, systems developer, data processing service provider, etc.) in different cultural settings were a part of the interview sample.

As a group, the 32 managers interviewed in both countries were well educated and had a good deal of experience. Almost all had bachelor's degrees in engineering, mathematics, or computer science. In Japan five had master's degrees: four of them in computer science, one in mechanical engineering. Three had degrees from US universities. In the United States one manager had a Phd in Computer Science and six had Master's degrees: one Masters of Business Administration, two in computer science, and four in an engineering discipline.

The number of years of experience of the managers ranged from 10 to 33 years. The average number of years for the managers in both countries was similar: in Japan 16 years compared to 18 years in the United States. Almost all of the Japanese managers had been with one employer for their entire career compared to the opposite for the US managers. That is, very few of the US managers had been with their present employer from the beginning of their working life.

The managers were from the following corporations:

United States - Agway Data Services, ATT Bell Laboratories, Eastman Kodak, General Electric (two divisions), and IBM (two divisions).

Japan - Hitachi (including a subsidiary Hitachi Software Engineering), IBM Japan, Japan Information Processing Service Co., KSC, Mitsubishi Electric Company, NEC (including a subsidiary NEC Software), Nihon Unisys, Software Research Associates, and Toshiba.

Context of Development

The context as described by the managers was important because of possible relationships amongst problems and these contextual variables. Managers were asked to characterize the development environment and software processes used, to describe the attributes of their projects, to present a profile of the personnel who worked on their development efforts, and to discuss the major constraints placed on their efforts.

Analysis of the data suggests that there were similarities and differences between the two countries surrounding the context in which software development took place, as summarized in Table 1.

The similarities were primarily technology related. It was found that Japanese software projects used analogous hardware/software systems and languages as the United States projects. Also, a typical Japanese software project used comparable software development tools as in United States projects. These findings are equivalent to the results of a study done by [Cus90]. However, ten years ago a survey of the state-of-practice in the US and Japan found that tool development and use was more widespread in Japan [ZYH⁺84].

Most of the tools in both countries were products of the United States. Japanese software factories have been widely publicized [TM84, Mat89, Cus89, Cus91, Cus92]. However, the Japanese managers interviewed in this study spoke rarely about the application of the software factory approach. They stated that the use of software factory methods were only applicable to well defined domains where there was a knowledge base of experience. This finding of limited application of the software factory approach is in contrast to reports that imply its widespread use [Cus91, Cus92]. Others have concluded that the software factory approach is not common in Japan [Mat91].

Comparative software processes guided the development projects, which consisted primarily of the Waterfall Model with some prototyping and interactive enhancement. Many of the managers felt that the existing models do not reflect what actually needs to be done and that they often must deviate from the company or government standard.

Project types and sizes were analogous between the two countries and included a wide variety of application domains. The average size ranged from 20,000 to 200,000 Lines of

Table 1: Software Development Context for United States and Japan

Contextual Variable	Representative Attributes
Development Environments	<p><i>SIMILAR</i></p> <ul style="list-style-type: none"> o Mainframe computers, Unix workstations, PCs <p>Languages - COBOL, C, C++, Fortran</p> <ul style="list-style-type: none"> o SW tools - design, CM, testing, tracking <p><i>JAPAN</i></p> <ul style="list-style-type: none"> o Distributed development
Software Process	<p><i>SIMILAR</i></p> <ul style="list-style-type: none"> o Normally Waterfall, company or client standards o Some prototyping and iterative enhancement o Need for improvement, especially for requirements
Project Types	<p><i>SIMILAR</i></p> <ul style="list-style-type: none"> o Information systems, Software tools, Manufacturing <p><i>UNITED STATES</i></p> <ul style="list-style-type: none"> o Real-time embedded systems
Project Size Ranges	<p><i>SIMILAR</i></p> <p>Average - 20,000 to 200,000 LOC</p> <ul style="list-style-type: none"> o Smallest - 500 LOC o Largest - 7,000,000 LOC
Personnel	<p><i>UNITED STATES</i></p> <ul style="list-style-type: none"> o Min. BS in Engineering or CS, many advanced degrees o Average experience, 5 to 15 yrs o Primarily in-house personnel <p><i>JAPAN</i></p> <ul style="list-style-type: none"> o Most had BS or BA degrees, some advanced degrees o Average experience, 2-5 years o Primarily subcontractor personnel
Constraints	<p><i>SIMILAR</i></p> <ul style="list-style-type: none"> o Primary - budget, time o Some - quality, customer imposed methods and tools <p><i>JAPAN</i></p> <ul style="list-style-type: none"> o Some had no budget constraint

Code (LOC).

The amount of experience and education of the personnel did not vary appreciably from one manager to another within the United States or within Japan, but there were differences between countries. The majority of the personnel in the United States had a minimum of a bachelor's degree, most often in computer science, engineering, or mathematics. There were also many personnel with advanced degrees. The experience of the software developers often averaged 5 to 15 years. In Japan, especially in the subsidiaries and software houses, the personnel did not have degrees, or their degrees were in business and the liberal arts. The average experience was most often 2 to 5 years.

One company in the United States had recently changed its' philosophy for hiring new software engineers. Previous to two years ago, they primarily hired new graduates with Master's degrees in the sciences, engineering, or mathematics. Presently they are finding it more cost effective to hire Associate degreed graduates for traditional programming tasks.

In the United States, the majority of the work was done by in-house personnel. In Japan, subcontracting was the norm.

Approximately half of the Japanese managers stated that they had few financial constraints placed on their projects - delivery and quality were more important. In the US, all managers mentioned fiscal constraints as being very important, in addition to schedule and quality.

Problems and Their Resolution

Some problems were similar. For example, managers in both countries experienced many problems in defining requirements and in resolving the volatility of the requirements. The differences between the two sets of managers encompassed non-technical or social issues related to the development problems encountered and the causes and resolutions of the problems.

Personnel Problems

The Japanese managers' biggest problem was the lack of experienced and skilled people. The number of computer science graduates needed greatly exceeded the supply, and the Japanese base of experienced software developers was less than in the United States. The managers often mentioned the lack of education, in addition to experience, as compared to the US. One manager said:

It is difficult because the programmers are not so educated as in the US. It depends on individual self development... People and education is a big problem. Maybe in the US there is a higher level to start. But in Japan the experience and education are very low.

Although the population as a whole is better educated than in the US in basic science and math, the computer science graduates are not as immediately useful as they are in the US ⁶.

⁶From an interview with Laszlo A. Belady in [Ale91].

Under normal conditions, the computer manufacturers in Japan maintain a steady hiring practice over the years. When this practice has not been followed, it has left managers with a shortage of personnel in some categories. As one of the Japanese managers stated:

Experienced people is a problem. We have people with under 10 years of experience. We have a group under age 32, but not many people over that. There are few people between 33 and 40. There is a gap. Why? Because of the oil crisis in the early 70s. We did not hire people then.

A constant hiring philosophy is more important in Japan than in the United States because of the common practice in Japan for an employee to stay with one company throughout ones career, especially in the larger companies. Also, these companies normally do not hire experienced people, therefore, the gap cannot be filled.

The lack of experienced people in Japan led to a proliferation of subcontracting amongst organizations which resulted in distributed development environments where project members were located in many different sites in Japan. This situation in turn led to other problems, such as breakdowns in communication between design teams and coding teams.

Only a few of the US managers complained about needing people. This problem, however, was not caused by the lack of the availability of people but because of constraints. These managers had to maintain a "fixed head count" in order to keep costs under control. That is, the managers were tasked to perform their development with only a certain amount of staff. In some cases they were allowed to subcontract out work but not hire new staff.

Situational Differences

The managers, when talking about development people, often described what they themselves did, what their people did, and/or philosophized about development people in general. The Japanese managers tended to explain ways in which they handled the shortage of experienced and skilled personnel. They often complained about the young people in general. Alternatively, the US managers rarely described what they themselves did with their people. Instead they recounted what their people did.

Three scenarios were defined for this study to help to understand this phenomenon, as depicted in Table 2. The first scenario covers those instances where managers, in both the United States and Japan, spoke about their concerns related to the changing world of software development. They expressed how some people fear change and need to be motivated to handle the changes. The managers provided for formal training on new technologies, counseled their people, and worked with them to help plan their career. Sometimes the managers felt that the responsibility for handing changes should be shared, as illustrated by the following:

We still deal with people. They have their baggage to deal with. There is a changing environment - the process technology and product technology. It is difficult getting all the people to keep up.

He was then asked what he did about that, and responded:

Education is part of it. It is a shared responsibility. People themselves must invest in themselves through incompany and outside classes, through journals, etc. People work with their managers to create a plan which is part of their performance evaluation. People have to continue to get better to stay even.

Part of it is ownership by the people. They have to own things, independent of whether it is forced on them or not.

The second scenario summarizes what has been said previously in this paper about the need for skilled and educated people. To help to alleviate this problem formal and on-the-job training was provided. The employees work was reviewed by others and, in some cases, was done by more experienced people. One Japanese manager said:

Most programmers have no experience and we need to educate them ... The expert works hard, instead of the low level developer.

While in Japan, preliminary results of this study were presented to a class of Computer Science students at Osaka University. One of the students remarked that he had heard about the uneven distribution of work and was concerned that he would be overworked, considering himself one of the better educated.

Many of the managers from Japan complained about the younger people and their lack of initiative and maturity. The following quote illustrates how one manager spoke about young people in general:

But young people want specifics. A general tendency in Japan is that young people want to be managed... The young people are not matured and cannot make a decision by themselves. We expect them to make a decision.

Scenario three shows a contrast to the complaints made by the Japanese managers. When the US managers spoke of their people, it was often-times complementary and not complaining. Overall the US managers were much more likely to mention technology or political issues rather than talk about people.

Requirements Problems

The requirements definition process was of great concern to both sets of managers. The managers had common objectives: to develop systems that met the customer's needs. They felt it was important to work closely with their customers throughout the development process, when possible.

There were complaints if they could not communicate directly with the customer, as stated by one manager:

We don't have direct interface with the customer. The systems people do - they get into that with the customer. There is filtering for what is going on. They may gloss over something. We don't know why they want something. We need to know so that we can make judgements.

Table 2: Situational Characteristics - People

Characteristic	Assumptions	Actions
CHANGING WORLD -Players	US and Japan	
	People fear change People need motivation People need ownership	Formal training Counseling Career planning
DEVELOPERS -Players	Japan	
	Lack of people Education limited People immature	On-the-job training Formal training Work reviewed, revised Others do work
DEVELOPERS -Players	US	
	People care, are committed People seek improvement People want realistic goals	No specific action

The data suggest that there were differences in attitude toward the approach to requirements' issues. The managers from Japan talked significantly more about collaborating and negotiating with the customers in defining requirements and when requirements changed - to work jointly with the customer in a spirit of cooperation. A partnership was implied, as illustrated by the following quote from one of the managers:

We make many documents for design. And if we have different points of view, we then discuss them with the customer. In Japan, you make the system with the customer, have a meeting every one, two, or three weeks. If we have any problems or questions, we discuss with them.

This is in contrast to US managers whose attitude was one of providing a service to the customer, knowing what was best for the customer. Prototypes and demonstrations were a form of protection from budget overruns and schedule delays.

Other Differences

Negotiations also played a role in subcontractor relations. Both sets of managers complained about poor subcontractor performance but how they tried to resolve these problems differed. The Japanese managers were more apt to treat their subcontractors as a part of the team and would work with these subcontractors to deliver the product to the customer. The United States managers spoke about how they could contractually force the subcontractor to deliver as originally planned.

The Japanese managers discussed how the Japanese people having difficulty making decisions. They said that this caused problems in two areas. First, the customers would not decide what they wanted, which resulted in incomplete and changing requirements. Secondly, project members would not make technical decisions, such as when designing a system. This led to project delays, and also, to frustration amongst project members.

Concluding Remarks

The above discussion summarizes the similarities and differences found in this study between the state-of-practice of software development in the United States and Japan. Examining the technological aspects of software development, there appears to be few distinguishing characteristics between the practices of the two countries. In contrast, the examination of management and sociological issues provides insight into the differences.

The concern for the lack of software engineers by the Japanese managers is not a new discovery [Cus91, HKG91]. The Ministry of International Trade and Industry (MITI) has estimated that by the year 2000, there will be a shortage of about 965,000 software engineers in Japan [sof90].

Historically the computer users in Japan preferred custom made software in order to meet their requirements.⁷ These attitudes are changing because of the increasing difficulties in finding qualified software personnel and the increasing costs of these personnel [sof90].

Since the interviews were performed in this study there has been a growth of economic problems in Japan which has affected the software industry. These economic problems, a growing concern for the lack of competitiveness of the software industry, and the acknowledgement of the inefficiency of custom software development have forecasters predicting a major change in the software industry in Japan. A recent newspaper article referenced industry executives who predicted that Japanese corporations will tend more toward packaged software and that software houses lacking capital may be purchased by US organizations [Pol92].

Acknowledgements

The author is deeply indebted to the managers interviewed for this study and to all those who helped in setting up the interviews. My visit to Japan was hosted by Koji Torii from Osaka University and Kouichi Kishida from Software Research Associates. Their hospitality, assistance and cooperation is very much appreciated. Also acknowledged is the New York State Center for Advanced Technology in Computer Applications and Software Engineering (CASE) Center for their support of this research.

References

- [Ale91] M. Alexander. Narrow is out, Interview with Laszlo A. Belady. *Computerworld Campus Edition*, 4(1):19-21, October 1991.

⁷The marketing research firm, Input, estimates that the percentage of packaged software costs versus total software costs is 20 percent for Japan, 60 percent for Europe, and 75 percent for the United States [Pol92].

- [CKI88] B. Curtis, H. Krasner, and N. Iscoe. A field study of the software design process for large systems. *Communications of the ACM*, 31(11):1268–1287, 1988.
- [CST89] Scaling Up: A Research Agenda for Software Engineering. Technical report, Computer Science and Technology Board (CSTB), National Research Council, 1989.
- [Cus89] M. A. Cusumano. The software factory: A historical interpretation. *IEEE Software*, 6(2):23–30, March 1989.
- [Cus90] M. A. Cusumano. A quantitative analysis of U.S. and Japanese practice and performance in software development. *Management Science*, 36(11):7–29, November 1990.
- [Cus91] M. A. Cusumano. *Japan's Software Factories: A Challenge to U.S. Management*. Oxford University Press, 1991.
- [Cus92] M. A. Cusumano. The Japanese Software Challenge. *Venture Japan, The Journal of Global Strategic Alliances*, 3(3):21–23, 1992.
- [GS67] B. Glaser and A. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing Co., 1967.
- [HKG91] W.S. Humphrey, D.H. Kitson, and J. Gale. A Comparison of US and Japanese Software Process Maturity. In *Proceedings of the International Conference on Software Engineering*, pages 38–49, 1991.
- [Kim83] K.H. Kim. A look at Japan's development of software engineering technology. *IEEE Computer*, 16(5):26–37, May 1983.
- [Mat89] Y Matsumoto. An overview of Japanese Software Factories. In *Japanese Perspectives in Software Engineering*, pages 303–320. Addison-Wesley Publishing Company, Inc., 1989.
- [Mat91] Tomoo Matsubara. Project Management in Japan. *American Programmer*, pages 41–50, June 1991.
- [Mye85] W. Myers. MCC:Planning the revolution in software. *IEEE Software*, pages 68–73, November 1985.
- [OB91] M. Ohba and V.R. Basili. A study of large scale software testing. *Proceedings of the Sixteenth Annual Software Engineering Workshop*, 1991.
- [Pat90] M.Q. Patton. *Qualitative Evaluation and Research Methods*. Sage, 1990.
- [Pol92] A. Pollack. Software - Japan's Little Nightmare-in-the-Making. *New York Times*, October 1992.
- [SB88] B.E. Swanson and C.M. Beath. The use of case study data in software management research. *The Journal of Systems and Software*, 8:63–71, 1988.

- [Sca89] W. Scacchi. Modeling software evolution: a knowledge-based approach. *ACM SIGSOFT Software Engineering Notes*, 14(4), June 1989.
- [sof90] 1990 Japanese Computer Software Market. Technical report, Software Information Center, Japan, 1990.
- [TM84] D. Tajima and T. Matsubara. Inside the Japanese Software Factory. *IEEE Computer*, 17(3):34–43, March 1984.
- [ZYH+84] M. Zelkowitz, R.T. Yeh, R.G. Hamlet, J.D. Gannon, and V.R. Basili. Software Engineering Practices in the US and Japan. *IEEE Computer*, 17(6):57–66, June 1984.

Insights into Software Development in Japan

Lorraine M. Duvall

Syracuse University
CASE Center
School of Information Studies

Osaka University

Duvall Computer Technologies, Inc.

OUTLINE

- **Motivation**
- **Study Profile**
- **Context Differences**
- **Problems and Resolutions**
- **Summary of Differences**

MOTIVATION

- Study how software is managed
 - From the managers perspective
 - Not how it *should* be managed
- Examine as a sociological phenomenon
 - context matters
 - studying problems provides insights
 - discovery versus prediction

SW INDUSTRY IN JAPAN

- Principal Players
 - Computer Manufacturers
 - Commercial Users
 - Subsidiaries of CM and Users
 - Independent SW Houses
- CM develop for Users
- Much subcontracting at all levels

TECHNOLOGY ATTRIBUTES

- **Use of tools and processes similar**
- **Tools were products of the US**
- **Software Factory approach not widely practiced**
 - **applicable to only well defined domains**

PERSONNEL - JAPAN

What managers said

- **Biggest problem we have**
 - **Not experienced**
 - **Lack initiative, are immature**
- **Problem causes other problems**
 - **Better people work hard**
 - **Subcontracting nightmare**
 - **Managers played the role of a parent**

SAMPLE PROFILE

- **14 companies**
 - **5 US, 7 Japanese**
 - **2 US - Japan affiliates**
- **32 managers**
- **Multiple development environments**
- **Diverse project types and sizes**

CONTEXT DIFFERENCES

JAPAN

- **Distributed development**
- **Personnel**
 - **Less experience**
 - **Fewer advanced degrees**
- **subcontracting the norm**

PERSONNEL - UNITED STATES

- **Seldom talked about what they, themselves did**
- **Never complained, usually complementary**
- **Managers played the role of a colleague**

REQUIREMENTS PROCESS

- **Big problem in both countries**
- **Managers knew how to move forward**
 - **close customer contact**
- **Attitude to customers different**

ATTITUDES TOWARDS CUSTOMERS

JAPANESE MANAGERS

- Worked jointly as partners
- Collaboration and negotiation important

US MANAGERS

- Service provider
- Agreement - a form of protection

SUBCONTRACTOR RELATIONS

JAPANESE MANAGERS

- Part of the team
- Negotiations important

US MANAGERS

- Contractually force performance

DECISION MAKING

- **Problem in Japan**
- **Customers - not decide what they wanted**
 - **Incomplete and changing requirements**
- **Project members - not make technical decisions**
 - **Project delays**
 - **Frustration**

CONCLUDING REMARKS

- **People, not technology, constitute the difference**
- **What is different - Roles managers play**
 - **Developers - Parent or Colleague**
 - **Customers - Collaborator or Service Provider**
 - **Subcontractors - Partner or Enforcer**

SOME PREDICTIONS
New York Times - October 11, 1992

- Software developers
 - Experiencing the economic squeeze
- Customers
 - Shift from custom to packaged software
- US companies
 - Purchase Japanese SW Houses

THE CULTURES ARE DIFFERENT

